

Reusable Components

The frame gallery contains a number of reusable components that can expand the functionality of the generated dialog. These components are objects which use different Natural object types.

This section describes the object types that are used for each component and how each are called.

- Communication with the Command Processor
 - Communication with the Data Buffer
 - Starting a Dialog (Application, Function, Listing)
 - Processing Status of Dialog Elements
 - Message Window
 - Date Validation
 - Numeric Validation
 - Natural Subprogram: ZXXNC00N
 - Logical Locking
-

Communication with the Command Processor

The control and manipulation of commands from application dialogs is performed by the command processor. Subroutines for communication with the command processor are defined in copy code ZXFXCP0C.

- Subroutine: Z_CMD_DISABLE
- Subroutine: Z_CMD_ENABLE
- Subroutine: Z_CMD_ADD_CTRL
- Subroutine: Z_SEND_CMD_PROC
- Operation: Z_CMD_CHECK
- Operation: Z_CMD_UNCHECK
- Operation: Z_CMD_DELETE
- Operation: Z_CMD_RENAME
- Operation: Z_CMD_REPLACE
- Operation: Z_CMD_DIL_REPLACE

Subroutine: Z_CMD_DISABLE

Description

Disables all dialog elements associated to a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command to be disabled

Example

```
MOVE 'Z MODIFY' TO LZ_COMMAND
PERFORM Z_CMD_DISABLE
```

Subroutine: Z_CMD_ENABLE

Description

Enables all dialog elements associated to a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command to be enabled

Example

```
MOVE 'Z MODIFY' TO LZ_COMMAND
PERFORM Z_CMD_ENABLE
```

Subroutine: Z_CMD_ADD_CTRL

Description

Assigns an additional dialog element (bitmap or push button) to a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_CONTROL	The handle of the dialog element
Input	LZ_COMMAND	The command to which the bitmap or push button is to be assigned. For a push button, the assignment can be omitted (command = value of the attribute COMMAND-ID)

Example

```
MOVE 'Z MODIFY' TO LZ_COMMAND
MOVE #BM-MODIFY TO LZ_CONTROL
PERFORM Z_CMD_ADD_CTRL

MOVE #PB-MODIFY TO LZ_CONTROL
PERFORM Z_CMD_ADD_CTRL
(attribute STRING from push button is 'Z MODIFY')
```

Subroutine: Z_SEND_CMD_PROC

Description

This subroutine is used to send all other requests to the command processor. The variable LZ_EVENT is used to indicate the processing to be performed.

Operation: Z_CMD_CHECK

Description

Places a check mark on a menu item associated to a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command

Example

```
MOVE 'Z_CMD_CHECK' TO LZ_EVENT  
MOVE 'DATE' TO LZ_COMMAND  
PERFORM Z_SEND_CMD_PROC
```

Operation: Z_CMD_UNCHECK

Description

Removes check mark on a menu item associated to a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command

Example

```
MOVE 'Z_CMD_UNCHECK' TO LZ_EVENT  
MOVE 'DATE' TO LZ_COMMAND  
PERFORM Z_SEND_CMD_PROC
```

Operation: Z_CMD_DELETE

Description

Deletes all dialog elements associated with a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command

Example

```
MOVE 'Z_CMD_DELETE' TO LZ_EVENT
MOVE 'SALARY' TO LZ_COMMAND
PERFORM Z_SEND_CMD_PROC
```

Operation: Z_CMD_RENAME

Description:

Assigns all dialog elements for a command to a new command.

The following attributes of the dialog element are also modified:

Menu Item:	STRING DIL-TEXT
Tool bar Item:	DIL-TEXT BITMAP-FILE-NAME
push button:	STRING DIL-TEXT
Bitmap:	DIL-TEXT BITMAP-FILE-NAME

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Old command
Input	LZ_COMMAND_NEW	New command

Example

```
MOVE 'Z_CMD_RENAME' TO LZ_EVENT
MOVE 'SALARY' TO LZ_COMMAND
MOVE 'VACATION' TO LZ_COMMAND_NEW
PERFORM Z_SEND_CMD_PROC
```

Operation: Z_CMD_REPLACE

Description:

Replaces place holder values for all dialog elements of a command name (attribute STRING).

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command
Input	LZ_COMMAND_FILL (1)	Value for place holder :1:
Input	LZ_COMMAND_FILL (2)	Value for place holder :2:
Input	LZ_COMMAND_FILL (3)	Value for place holder :3:

Example

```
MOVE 'Z_CMD_REPLACE' TO LZ_EVENT
MOVE 'CALCULATE' TO LZ_COMMAND
MOVE 'VACDAYS' TO LZ_FILL_TXT (1)
PERFORM Z_SEND_CMD_PROC
```

Operation: Z_CMD_DIL_REPLACE

Description

Replaces place holder values in DIL-text for all dialog elements of a command.

Parameters

Contained in the local data area ZXXSTD0L

Input/Output	Parameter Variable	Description
Input	LZ_COMMAND	Command
Input	LZ_COMMAND_FILL (1)	Value for place holder :1:
Input	LZ_COMMAND_FILL (2)	Value for place holder :2:
Input	LZ_COMMAND_FILL (3)	Value for place holder :3:

Example

```
MOVE 'Z_CMD_DIL_REPLACE' TO LZ_EVENT
MOVE 'CALCULATE' TO LZ_COMMAND
MOVE 'VACDAYS' TO LZ_FILL_TXT (1)
PERFORM Z_SEND_CMD_PROC
```

Communication with the Data Buffer

Global data from the data buffer are contained in the local data area XXGLOBL. A copy of this local data area is available as a local data area for each application dialog. Subroutines for communication with the data buffer are defined in the copy code ZXFXCD0C.

- Natural Subroutine: Z_GIVE_GLOBAL
- Natural Subroutine: Z_UPDATE_GLOBAL

Natural Subroutine: Z_GIVE_GLOBAL

Description

Requests global data from the data buffer. The current data are provided as a local copy in the local data area ZXXGLOBL.

Example

```
PERFORM Z_GIVE_GLOBAL
```

Natural Subroutine: Z_UPDATE_GLOBAL

Description

The global data is updated in the data buffer and then is distributed to all dialogs of the application. The variable contents of the local copy of the local data area ZXXGLOBL are transferred to the data buffer.

Example

```
MOVE 'Unknown' TO LZ_GLOBAL.LZ_UNTITLED  
PERFORM Z_UPDATE_GLOBAL
```

Starting a Dialog (Application, Function, Browse)

External Subroutine: Z_INVOKE_FUNCTION

This subroutine is used to start an application, a function or a browse function within the application.

A function can be started directly using a Function ID or via the combination of Action and Object Type.

Parameters

Group PZ_LOCAL of the parameter data area ZXXLOC0A

Only those parameters which are relevant for the call to a subroutine (reading or writing) are described. Variables not listed below should not be modified in that they contain parameter data area values which are required by each dialog.

Input/Output	Parameter Variable	Description
Input	PZ_CMD_ID	Command to be executed
Input	PZ_CMD_TYPE_MAIN	Command main type
Input	PZ_ACT_TYPE_CUR	Command sub type
Input	PZ_CMD_PARM	Command parameter
Input	PZ_SEL_KEY	Key for dialog to be started
Input	PZ_KEY_FILLED	TRUE key is filled FALSE key is not filled
Output	PZ_DLG_NAME	Natural name of the dialog
Output	PZ_DLG_ID	Natural ID of the dialog
Output	PZ_RSP	Return message value

The parameter PZ_SEL_KEY should only be completed if PZ_KEY_FILLED is set to TRUE. This is only meaningful when a function or a listing is to be started.

The following sections describe how the parameter variables are to be filled.

Application Start via Command

Parameter	Value
PZ_CMD_ID	Command ID, Command must be of the type 'Start an Application'. Command parameter indicates the Application to be started.

Example

```
RESET PZ_LOCAL.PZ_CMD_GROUP
MOVE 'Z_TOP' TO PZ_LOCAL.PZ_CMD_ID
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT
```

Application Start without Command

Parameter	Value
PZ_CMD_TYPE_MAIN	LZ_CMD_TYPE_APPL (constant from local data area ZXX00CL)
PZ_CMD_PARM	Application ID

Example

```

RESET PZ_LOCAL.PZ_CMD_GROUP
MOVE LZ_CMD_TYPE_APPL TO PZ_LOCAL.PZ_CMD_TYPE_MAIN
MOVE 'Z_TOP' TO PZ_LOCAL.PZ_CMD_PARM
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT

```

Start Browse via Command

Parameter	Value
PZ_CMD_ID	Command ID, Command must be of type 'Start Browse'. Command parameter indicates the object type for which the listing is to be started.

Example

```

RESET PZ_LOCAL.PZ_CMD_GROUP
MOVE 'LZ_EMPLOY' TO PZ_LOCAL.PZ_CMD_ID
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT

```

Start Browse without Command

Parameter	Value
PZ_CMD_TYPE_MAIN	LZ_CMD_TYPE_OBJ (constant from local data area ZXX00CL)
PZ_CMD_PARM	Object type ID for which the browse is to be started

Example

```

RESET PZ_LOCAL.PZ_CMD_GROUP
MOVE LZ_CMD_TYPE_OBJ TO PZ_LOCAL.PZ_CMD_TYPE_MAIN
MOVE 'EMPLOYEE' TO PZ_LOCAL.PZ_CMD_PARM
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT

```

Function Start via Command

Parameter	Value
PZ_CMD_ID	Command ID, Command must be of type 'Start a Function'. Command parameter indicates the function to be started.
PZ_SEL_KEY	Key value
PZ_KEY_FILLED	TRUE if key value is submitted

Example

```

RESET  PZ_LOCAL.PZ_CMD_GROUP
      PZ_LOCAL.PZ_SEL_KEY
      PZ_LOCAL.PZ_KEY_FILLED
MOVE   'ADM-EMP'    TO PZ_LOCAL.PZ_CMD_ID
MOVE   '4711'        TO PZ_LOCAL.PZ_SEL_KEY
MOVE   TRUE          TO PZ_LOCAL.PZ_KEY_FILLED
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT

```

Function Start without Command

Parameter	Value
PZ_CMD_TYPE_MAIN	LZ_CMD_TYPE_FCT (constant from local data area ZXX000CL)
PZ_ACT_TYPE_CUR	Action Type (see local data area ZXX000CL)
PZ_CMD_PARM	Function ID
PZ_SEL_KEY	Key value
PZ_KEY_FILLED	TRUE if key value is submitted

Example

```

RESET  PZ_LOCAL.PZ_CMD_GROUP
      PZ_LOCAL.PZ_SEL_KEY
      PZ_LOCAL.PZ_KEY_FILLED
MOVE   LZ_CMD_TYPE_FCT TO PZ_LOCAL.PZ_CMD_TYPE_MAIN
MOVE   LZ_CMD_TYPE_ACT_DEL TO PZ_LOCAL.PZ_ACT_TYPE_CUR
MOVE   'ADM-EMP'    TO PZ_LOCAL.PZ_CMD_PARM
MOVE   '4711'        TO PZ_LOCAL.PZ_SEL_KEY
MOVE   TRUE          TO PZ_LOCAL.PZ_KEY_FILLED
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT

```

Function Start via Action Change

Parameter	Value
PZ_CMD_ID	Command ID, Command must be of type action The combination of Action and current object type indicates the function to be started.
PZ_SEL_KEY	Key value
PZ_KEY_FILLED	TRUE if key value is submitted

Example

```
RESET  PZ_LOCAL.PZ_CMD_GROUP
      PZ_LOCAL.PZ_SEL_KEY
      PZ_LOCAL.PZ_KEY_FILLED
MOVE   'Z MODIFY'  TO PZ_LOCAL.PZ_CMD_ID
MOVE   '4711'       TO PZ_LOCAL.PZ_SEL_KEY
MOVE   TRUE         TO PZ_LOCAL.PZ_KEY_FILLED
PERFORM Z_INVOKE_FUNCTION PZ_LOCAL
#DLG$PARENT
```

Processing Status of Dialog Elements

The frames control the data modification of a function and disable dialog elements in display functions. The frame controls, by default, all dialog elements which CHANGE events are not suppressed.

The following subprograms may be used to force the frame to control dialog elements in any other way.

- Natural Subprogram: ZXXCTIGN
- Natural Subprogram: ZXXCTKYN
- Natural Subprogram: ZXXCTMON
- Natural Subroutine: Z_DIALOG_MODIFIED

Natural Subprogram: ZXXCTIGN

Parameter: HANDLE OF ANY

Description

This subprogram can be used to force the frame to ignore a dialog element, e.g. containing a start value or controlling the display mode.

Parameters

Input/Output	Parameter Variable	Description
Input	HANDLE OF ANY	Handle of dialog element

Example

```
CALLNAT 'ZXXCTIGN' #TB-DISPLAY_ADDITIONAL_INFO
```

Natural Subprogram: ZXXCTKYN

Parameter: HANDLE OF ANY

Description

This subprogram can be used to mark a dialog element as a key component. The frame interprets any modification as a modification of the key value. (Only effective in a mass processing dialog).

Parameters

Input/Output	Parameter Variable	Description
Input	HANDLE OF ANY	Handle of dialog element

Example

```
CALLNAT 'ZXXCTKYN' #IF-CUSTOMER_ID
```

Natural Subprogram: ZXXCTMON

Parameter: HANDLE OF ANY

Description

This subprogram can be used to mark a dialog element as a dialog element. The frame interprets any modification as a modification of the data to be processed.

Parameters

Input/Output	Parameter Variable	Description
Input	HANDLE OF ANY	Handle of dialog element

Example

```
CALLNAT 'ZXXCTMON' #SB-CUSTOMER_TYPE
```

Natural Subroutine: Z_DIALOG_MODIFIED

Description

Forces dialog status to "modified" in the following frames: maintain, subdialog, modal windows, and nonstandard dialog.

Parameters

None.

Example

```
PERFORM Z_DIALOG_MODIFIED
```

Message Window

With Windows applications, messages are usually provided in a message window or a status line. The copy code ZXFXMSGC defines a subroutine which is used to control messages. Message texts are stored in a Natural message file. Multiple language support as well as modification of messages independent of program logic is thereby possible.

A message text is constructed as follows:

```
text:1:text:2:text:3:text
```

The place holders :1:, :2: and :3: are set dynamically during runtime.

A place holder can occur multiple times in a single message.

A new line within a message is created by entering two colons in succession (::).

Natural Subroutine: Z_DISPLAY_MESSAGE

Description

This subroutine is used to display messages in a message window or in a status line of the current window. Up to 3 message texts (one line per text) can be displayed at the same time in a message window.

Parameters

Group PZ_MSG of the parameter data area ZXXMSG0A

Input/Output	Parameter Variable	Description
Input	PZ_MSG_TYPE	Message type (see below)
Input	PZ_MSG_NUM (3 occurrences)	Message numbers
Input	PZ_MSG_FILL (3 occurrences)	Place holders
Input	PZ_MSG_TITLE	Optional title for message window
Output	PZ_MSG_BUTTON	Active buttons for message window
Input	PZ_MSG_DLG (not part of the group)	Handle of parent window Usually #DLG\$WINDOW)

The following message types are defined as constants in the local data area ZXX000CL:

Constant	Description
LZ_MSG_BUTTON_OK	OK button
LZ_MSG_BUTTON_YES	YES button
LZ_MSG_BUTTON_NO	NO button
LZ_MSG_BUTTON_CANCEL	CANCEL button

Example

Message Text 1500: :1: employees live in :2:
--

```

MOVE 'BERLIN' TO #CITY
CITY_FIND.
FIND NUMBER EMPLOYEES WITH CITY = #CITY

MOVE LZ_MSG_TYPE_INFO TO PZ_MSG_TYPE
MOVE 1500 TO PZ_MSG_NUM(1)
MOVE *NUMBER(CITY-FIND.) TO PZ_MSG_FILL(1)
MOVE #CITY TO PZ_MSG_FILL(2)
PERFORM Z_DISPLAY_MESSAGE

```

Date Validation

Natural Subprogram: ZXXDATEN

Parameter Data Area: ZXXDATEA

Description

The subprogram verifies that the input date is a valid date. The input date can be in any of the Natural formats A10, A8, N8 or D. These formats are all available for the output date following a successful validation.

An alphanumeric date (A10 and A8) is checked according to the current date format.

The following rules apply for input dates of formats A8 and A10:

- the year can be omitted
- the century can be omitted when specifying the year
- leading zeros can be omitted for day and month entries

Parameters

Group PZ_DATE of the parameter data area ZXXDATEA

Input/Output	Parameter Variable	Description
Input	PZ_DATE_FORMAT	Date format for alphanumeric input and output date (see below) (default LZ_FORMAT_DTFORM)
Input	PZ_DATE_CONVERT	Variable format of input date (see below)
Input	PZ_DATE_IGNORE_DELIM	Only for alphanumeric input date FALSE: only the seperation character of date format is permitted (default) TRUE: any character is permitted as separation character
Input	PZ_DATE_CENTURY_LIMIT	Only for alphanumeric input date Century limit for 2 position century input (default = 0) Year. < Century Limit = 20th C. Year. >= Century Limit = Current C.
Input/Output	PZ_DATE_A10	Date using format A10
Input/Output	PZ_DATE_A8	Date using format A8
Input/Output	PZ_DATE_N	Date using format N8 (YYYYMMDD)
Input/Output	PZ_DATE_D	Date using format D
Input/Output	PZ_DATE_RSP	Response Code 0: Date OK 1: Date invalid 2: PZ_DATE_FORMAT invalid 3: PZ_DATE_CONVERT invalid

The valid values for PZ_DATE_FORMAT are defined as constants in the local data area ZXXDATCL.

Constant	Representation A8	Representation A10
LZ_FORMAT_US	MM/DD/YY	MM/DD/YYYY
LZ_FORMAT_GERMAN	DD.MM.YY	DD.MM.YYYY
LZ_FORMAT_EUROPE	DD/MM/YY	DD/MM/YYYY
LZ_FORMAT_INTERNATIONAL	YY-MM-DD	YYYY-MM-DD
LZ_FORMAT_DTFORM	according to Natural parameter DTFORM	according to Natural parameter DTFORM

The valid values for PZ_DATE_CONVERT are defined as constants in the local data area ZXXDATCL.

Constant	Variable Format for Input Date
LZ_CONVERT_A8	A8
LZ_CONVERT_A10	A10
LZ_CONVERT_N	N8
LZ_CONVERT_D	D

Example

```

MOVE LZ_FORMAT_GERMAN    TO PZ_DATE_FORMAT
MOVE LZ_CONVERT_A10      TO PZ_DATE_CONVERT
MOVE '1.3.95'            TO PZ_DATE_A10

CALLNAT 'ZXXDATEN'    PZ_DATE

```

Output

PZ_DATE_A10	01.03.1995
PZ_DATE_A8	01.03.95
PZ_DATE_N	19950301
PZ_DATE_D	Natural internal date format

Numeric Validation

Function

To validate and convert numerical values.

Name:	ZXXNC00N
PDA:	ZXXNC00A

Natural Subprogram: ZXXNC00N

Parameter Data Area: ZXXNC00A

Description

The subprogram validates and converts numerical values.

Parameters

Group PZ_NC_PARMS of the parameter data area ZXXNC00A

Input/Output	Parameter Variable	Description
Input	PZ_NC_OP	Operation code
Input/Output	PZ_NC_VALUE	Value to be processed
Input/Output	PZ_NC_BEFORE_DC	Length before the decimal character (maximum 27)
Input/Output	PZ_NC_AFTER_DC	Length after the decimal character (maximum 27)
Input	PZ_NC_DC	Decimal character
Input	PZ_NC_RSP	Response codes

The following operations are defined in local data area ZXXNC0CL:

Operation	Description
LZ_NC_CHECK	Checks the numerical input value. The value can contain a maximum of one decimal character.
LZ_NC_GIVE_LEN	Supplies the length of the input value (both before and after the decimal character).
LZ_NC_EXACT_LEN	Checks the input value against the corresponding defined length.
LZ_NC_LEN_NO_ZERO	Checks the input value against the corresponding defined length, whereby no leading zeros are allowed.
LZ_NC_FILL.LEADING	Fills the input value according to the definition with leading zeros.
LZ_NC_FILL.FOLLOWING	Fills the input value according to the definition with trailing zeros (after the decimal character).
LZ_NC_FILL	Fills the input value according to the definition with leading zeros and trailing zeros (after the decimal character).

Example

```

RESET PZ_NC
MOVE LZ_NC_CHECK    TO PZ_NC_OP
MOVE '27'           TO PZ_NC_VALUE
MOVE '3'            TO PZ_NC_LEN_BEFORE_DC

CALLNAT 'ZXXNC00N'  PZ_NC_PARMS

```

Logical Locking

Natural Subroutine: Z_CHECK_AND_LOCK_RECORD

Description

Determines if a key or key area is locked by another transaction, and, if possible, locks the key or key area.

Parameters

The parameters are contained in the local data area ZXXSTD0L.

Input/Output	Parameter Variable	Description
Input	LZ_LOCK_OBJ_ID	Identifier of the object type
Input	LZ_LOCK_KEY	The key to be locked or the beginning of the key area to be locked
Input	LZ_LOCK_KEY_END	The end of the key area to be locked
Output	LZ_VAL_ERR	TRUE: Locking is not possible

If LZ_VAL_ERR has a setting of TRUE, the subroutine also inserts the corresponding error number into the variable PZ_MSG.PZ_MSG_NUM(1).

Example

```

MOVE  'ARTICLE'          TO LZ_LOCK_OBJ_ID
MOVE  PZ_LOCAL.PZ_KEY   TO LZ_LOCK_KEY

PERFORM Z_CHECK_AND_LOCK_RECORD

IF LZ_VAL_ERR
  BACKOUT TRANSACTION
  PERFORM Z_DISPLAY_MESSAGE
ELSE
  END TRANSACTION
END-IF

```